

0
P1

200 RPT AM OUT
202 RPT CHANGE CHANNEL 0
205 PORE 16514.0
210 PAST
215 LAT L=USR 10002
217 RPT R 15 SUPN SCALING
FACTOR
219 LAT X=1
220 LAT O=X+PARK 10015
225 SLUM
230 DUSUN 130
240 GOTO MENU
END

Fig. 3(b)
BASIC LAYOUT
FOR A/D CONVERTER

```

130: INPUT D
131: PRINT D
132: REM NOW MAKE CURRENT D TO 2 SEP
  ANGLE DEGREE
133: LET PRINT (D/100)
137: LET TELL (10-INT(D/100)/10)
139: LET UNIT (D-INT(D/100)-(D/100)
140: REM NOW CHANGE UNIT TO AN
  SPECIAL CODE
142: LET H=10
144: LET T=100
146: LET UNIT=10
148: REM COULD HAVE BEEN DONE IN
  145 INSTEAD
150: DONE SETUP, H
152: DONE SETUP, T
154: DONE SETUP, D
156: REM CALL LEAD, T
158: FAST
160: LEFT-HSA LEDOUT
162: STOP
164: RETURN

```

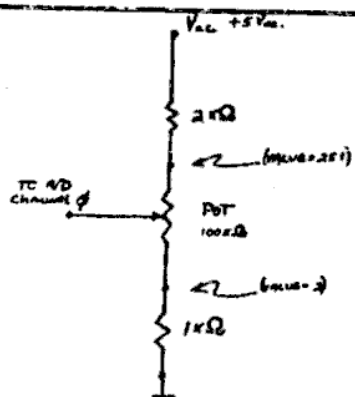
Fig. 3(a)
LEAD

```

1 REM *****PROGRAM*****
2 THIS IS A MARK AREA
3 FAST
4 REM BASIL LEDOUT
5 LET H=0
6 LET T=0
7 LET U=0
8 LET OUTPUT=10000
9 LET (M=41000
10 DIM V(4)
11 REM MAIN
12 SLW
13 INPUT "INPUT DECIMAL VALUE"
14 INPUT D
15 PRINT D
16 LET M=4-INT (D/100)
17 LET T=732-INT (D-(M*4))=100
18
19 LET U=1+INT (D-(M*4))=100-
20 (732+10)
21
22 LET V(1)=H
23 LET V(2)=T
24 LET V(3)=U
25 LET V(4)=0
26 FOR J=1 TO 50
27 FOR I=1 TO 3
28 POK (OUT,V(I))
29 FAST
30 LET A=U+8 OUTPUT
31 NEXT I
32 NEXT J
33 FONE Q=V(4)
34 LET I=U+8 OUTPUT
35 PRINT "AS=INT"
36 INPUT AS
37 IF AS=V THEN GOTO 100
38 IF AS=D THEN GOTO 200
39 STOP
40 REM BLADOUT
41 POK 16510
42 FAST
43
44 LET L=U+8 10002
45 LET U=U+8 16515
46 SLW
47 GOTO 115
48 STOP

```

Fig. 4
Pure BASIC LAYOUT
(v/ A-to-D)



```

3000 REM ADTEST
6000 FAST
6010 FOR I=1 TO 10
6020 MAKE I$=I*10
6030 LET L=USR 1000Z
6040 PRINT "VALUE=";P=2*I I$
6050 GOTO 6010
6060 RETURN

```

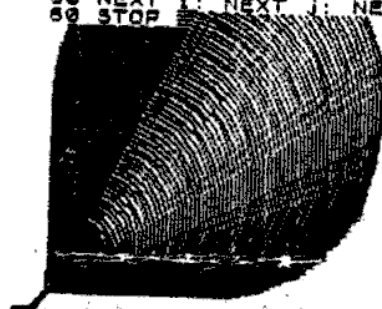
8. *How do you feel about the way the situation is being handled?*

VML_12 = 85
VML_12 = 201
VML_12 = 226
VML_12 = 189
VML_12 = 130
VML_12 = 406
VML_12 = 75
VML_12 = 246
VML_12 =
VML_12 =

```

10 FOR I=1 TO 10
20 FOR J=1 TO 10
30 FOR K=1 TO 100
40 CIRCLE I,J,K
50 NEXT K: NEXT J: NEXT I
60 STOP

```



What's wrong with this program

SAC/MSB/NSA/IR 7740

If your IBM keyboard starts acting up, it may be because of jump contacts between the keyboard's flat ribbon conductor and the main frame. I just adjusted 8 keys on the right hand side recently for no apparent reason. A quick look at the IBM schematic shows these keys (B,B,M...touch) to be controlled by the 28-A15 line on the B connector block. Apparently, the key's thin metal film on the plastic carrier can be warped or off during assembly.

Pull the ribbon out of their plugs and hold the ends up to a light. If the metal film was scratched off, you'll see daylight through the screen. Repair is simple, just take a sharp ^{pair of} ~~sharp~~ scissors and trim evenly 1/8" from the entire bottom edge of the flat cable (don't forget the neck).

Diagnosing the problem is just as easy - If you loose columns (e.g., V,C,P,S and 4, 1,2,3) you've lost contact with one of the XDR lines (in this case XDR1) on the 5 pin connector. If you've lost (as to my canal) you'll know 5 says, so suspect the right pin lines.

Sample extracted from VU-12



WHAT A CHARACTER

The Game Changer Interface (reviewed elsewhere) enables you to upload ATARI VCS (2600) cartridge contents into your EX/TS, SAVE them on tape and play 'em, using the EX/TS RAM. Since the copy of the game is in RAM, you can also change any part you wish, to suit your own personal style. The ATARI uses a 6507 (processor) significant changes require programming in 6502 machine code.

However, an understanding of 6502 machine code (MC) is not really necessary for you to change some aspects of your recorded ATARI games. Specifically, any character or icon (a symbol representing you, an object, enemy etc.) can be changed to another shape, just by FORKING new values into its data array.

In order to be represented on the screen, an object must have "bit images" stored somewhere in memory. The game software sends these bits to the hardware which in turn produces the picture you see on the screen. Here's a simple example of a face using an 8 X 8 bit image (that is, 8 bits wide by 8 bits high):

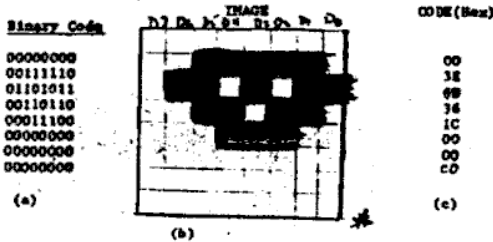


Fig. 1

Figure 1(a) gives the binary representation of our "face". Here ones (1's) will represent "turned on" bits and will be dark, while 0's are turned off or blank (the background color will show through). In Fig. 1(c), since this image is exactly 8 bits wide, we've shown the hexidecimal numbers representing one bit byte. This face can be stored in 8 consecutive memory locations.

If we were to pull up the ATARI game code on the screen a byte at a time and represent it as a series of black and white squares we could "see" the actual character images used in the game. Fig. 2 gives a partial program listing that will do just that.

Before you GOTO 7000 however, you may have to do a little detective work. A typical ATARI cartridge contains a 4K ROM which you will already have downloaded to the 4K of RAM starting at 16314. The 6502 interrupts (where the program restarts after something important or bad happens) point to the very high memory locations and addresses pointing to the actual program code will be stored there. You can use the disassembler or monitor to check the top sections of RAM (FFFF through FFFF) for these pointers. You can be reasonably sure the data tables for the characters are not in the immediate areas pointed to by those three sets of 2 bytes and should look elsewhere. Places to look are: 1) boundaries between pages (7000, 7300 etc. - a "page" is 256 bytes); 2) the beginning of the program; 3) addresses which occur frequently.

For ATARI BASIC we used the disassembler to spot check some of these areas and found absolute garbage at the beginning of the program. A good hard look at the 1 ROM (see Fig. 3) also shows the experienced programmer something which just about has to be nonsense. The recurrence of the same code number several times in succession is very unlikely in an MC program. These clues led us to choose 16314 as a starting address. After offset, this address corresponds to 7000(X) in the ATARI.

After going to 7000 we obtained the listing in Fig. 4. A pattern of dots is clearly emerging, but hard to identify as letters or figures. Connecting the dots (or using the inverse space instead of an asterisk) can help make the patterns more intelligible. We've done that for the "GOTO" command of ATARI BASIC. You should be able to pick out some other upside-down commands like PRINT, and ELSE. The game software and hardware automatically turn these right side up.

How does the routine work? First we dimension some new variables and pick a starting address. These are:

- S = Starting address (decimal only)
- D(1) = The binary 1 th Digit of a Byte
- A\$ = A string which will print either filled-in or MY square
- F = The decimal value of the byte at location S
- N = A temporary "port" of the F variable as it is reduced to its D or binary digits.

Next we PEEK the data stored in the currently chosen memory location. The subroutine at 8000 repeatedly divides the decimal number by 2, saving the remainder as D(J) to convert the decimal number to its binary (actually binary still stored as decimal) bit equivalents.

Returning to 7025ff from the subroutine we convert each "character" in the string A\$ to its bit pattern; it remains a blank if D(J)≠1 (that is, if its 0) and becomes an asterisk if D(J)=1.

Finally we print the memory location, decimal value and our now blocked out string of "bits". Don't forget that the actual addresses where this data table reside are some 12000 (decimal) memory locations higher in the ATARI. Also, the conventions used by this programmer may not be used in other games. In ATARI BASIC, in fact, two more, different, techniques are used for addressing the numbers (for NAME) and the display area titles.

You should be able to add these lines to the "change" program with no problems on 4K or less games. You could even just pick an address at random and observe the screen for intelligible patterns. For 8K games create a free standing

version of the program addition and try to PEEK the (with the actual cartridge installed) memory locations above 18000. Remember, the bank switcher may get in the way.

Some hints as to when to change in ATARI BASIC:

- 1) The symbol table for the display areas (STATUS, PROGRAM etc.) Abbreviating these would free up RAM for your own 6502 MC routines.
- 2) Change the "Block" graphics character to more interesting pictures (e.g., a silly face).
- 3) Change the alpha or numeric characters to provide a more interesting display. The letters, for example, can be converted to those of another language.
- 4) Your own ideas.

```

7000:BIN B(8)
7001:BIN A(4)
7002:PRINT " INPUT STARTING ADDR"
7003:
7004:INPUT S
7005:FOR I=1 TO 8:255
7006:LET F=PEEK S
7007:GOSUB 8000
7008:FOR J=1 TO 8
7009:LET D(J)=F/2
7010:IF D(J)=1 THEN LET A(J)=*
7011:
7012:PRINT I:
7013:PRINT A$
7014:
7015:GOTO 7005
7016:
7017:
7018:
7019:
7020:
7021:
7022:
7023:
7024:
7025:
7026:
7027:
7028:
7029:
7030:
7031:
7032:
7033:
7034:
7035:
7036:
7037:
7038:
7039:
7040:
7041:
7042:
7043:
7044:
7045:
7046:
7047:
7048:
7049:
7050:
7051:
7052:
7053:
7054:
7055:
7056:
7057:
7058:
7059:
7060:
7061:
7062:
7063:
7064:
7065:
7066:
7067:
7068:
7069:
7070:
7071:
7072:
7073:
7074:
7075:
7076:
7077:
7078:
7079:
7080:
7081:
7082:
7083:
7084:
7085:
7086:
7087:
7088:
7089:
7090:
7091:
7092:
7093:
7094:
7095:
7096:
7097:
7098:
7099:
7100:
7101:
7102:
7103:
7104:
7105:
7106:
7107:
7108:
7109:
7110:
7111:
7112:
7113:
7114:
7115:
7116:
7117:
7118:
7119:
7120:
7121:
7122:
7123:
7124:
7125:
7126:
7127:
7128:
7129:
7130:
7131:
7132:
7133:
7134:
7135:
7136:
7137:
7138:
7139:
7140:
7141:
7142:
7143:
7144:
7145:
7146:
7147:
7148:
7149:
7150:
7151:
7152:
7153:
7154:
7155:
7156:
7157:
7158:
7159:
7160:
7161:
7162:
7163:
7164:
7165:
7166:
7167:
7168:
7169:
7170:
7171:
7172:
7173:
7174:
7175:
7176:
7177:
7178:
7179:
7180:
7181:
7182:
7183:
7184:
7185:
7186:
7187:
7188:
7189:
7190:
7191:
7192:
7193:
7194:
7195:
7196:
7197:
7198:
7199:
7200:
7201:
7202:
7203:
7204:
7205:
7206:
7207:
7208:
7209:
7210:
7211:
7212:
7213:
7214:
7215:
7216:
7217:
7218:
7219:
7220:
7221:
7222:
7223:
7224:
7225:
7226:
7227:
7228:
7229:
7230:
7231:
7232:
7233:
7234:
7235:
7236:
7237:
7238:
7239:
7240:
7241:
7242:
7243:
7244:
7245:
7246:
7247:
7248:
7249:
7250:
7251:
7252:
7253:
7254:
7255:
7256:
7257:
7258:
7259:
7260:
7261:
7262:
7263:
7264:
7265:
7266:
7267:
7268:
7269:
7270:
7271:
7272:
7273:
7274:
7275:
7276:
7277:
7278:
7279:
7280:
7281:
7282:
7283:
7284:
7285:
7286:
7287:
7288:
7289:
7290:
7291:
7292:
7293:
7294:
7295:
7296:
7297:
7298:
7299:
7300:
7301:
7302:
7303:
7304:
7305:
7306:
7307:
7308:
7309:
7310:
7311:
7312:
7313:
7314:
7315:
7316:
7317:
7318:
7319:
7320:
7321:
7322:
7323:
7324:
7325:
7326:
7327:
7328:
7329:
7330:
7331:
7332:
7333:
7334:
7335:
7336:
7337:
7338:
7339:
7340:
7341:
7342:
7343:
7344:
7345:
7346:
7347:
7348:
7349:
7350:
7351:
7352:
7353:
7354:
7355:
7356:
7357:
7358:
7359:
7360:
7361:
7362:
7363:
7364:
7365:
7366:
7367:
7368:
7369:
7370:
7371:
7372:
7373:
7374:
7375:
7376:
7377:
7378:
7379:
7380:
7381:
7382:
7383:
7384:
7385:
7386:
7387:
7388:
7389:
7390:
7391:
7392:
7393:
7394:
7395:
7396:
7397:
7398:
7399:
7400:
7401:
7402:
7403:
7404:
7405:
7406:
7407:
7408:
7409:
7410:
7411:
7412:
7413:
7414:
7415:
7416:
7417:
7418:
7419:
7420:
7421:
7422:
7423:
7424:
7425:
7426:
7427:
7428:
7429:
7430:
7431:
7432:
7433:
7434:
7435:
7436:
7437:
7438:
7439:
7440:
7441:
7442:
7443:
7444:
7445:
7446:
7447:
7448:
7449:
7450:
7451:
7452:
7453:
7454:
7455:
7456:
7457:
7458:
7459:
7460:
7461:
7462:
7463:
7464:
7465:
7466:
7467:
7468:
7469:
7470:
7471:
7472:
7473:
7474:
7475:
7476:
7477:
7478:
7479:
7480:
7481:
7482:
7483:
7484:
7485:
7486:
7487:
7488:
7489:
7490:
7491:
7492:
7493:
7494:
7495:
7496:
7497:
7498:
7499:
7500:
7501:
7502:
7503:
7504:
7505:
7506:
7507:
7508:
7509:
7510:
7511:
7512:
7513:
7514:
7515:
7516:
7517:
7518:
7519:
7520:
7521:
7522:
7523:
7524:
7525:
7526:
7527:
7528:
7529:
7530:
7531:
7532:
7533:
7534:
7535:
7536:
7537:
7538:
7539:
7540:
7541:
7542:
7543:
7544:
7545:
7546:
7547:
7548:
7549:
7550:
7551:
7552:
7553:
7554:
7555:
7556:
7557:
7558:
7559:
7560:
7561:
7562:
7563:
7564:
7565:
7566:
7567:
7568:
7569:
7570:
7571:
7572:
7573:
7574:
7575:
7576:
7577:
7578:
7579:
7580:
7581:
7582:
7583:
7584:
7585:
7586:
7587:
7588:
7589:
7590:
7591:
7592:
7593:
7594:
7595:
7596:
7597:
7598:
7599:
7600:
7601:
7602:
7603:
7604:
7605:
7606:
7607:
7608:
7609:
7610:
7611:
7612:
7613:
7614:
7615:
7616:
7617:
7618:
7619:
7620:
7621:
7622:
7623:
7624:
7625:
7626:
7627:
7628:
7629:
7630:
7631:
7632:
7633:
7634:
7635:
7636:
7637:
7638:
7639:
7640:
7641:
7642:
7643:
7644:
7645:
7646:
7647:
7648:
7649:
7650:
7651:
7652:
7653:
7654:
7655:
7656:
7657:
7658:
7659:
7660:
7661:
7662:
7663:
7664:
7665:
7666:
7667:
7668:
7669:
7670:
7671:
7672:
7673:
7674:
7675:
7676:
7677:
7678:
7679:
7680:
7681:
7682:
7683:
7684:
7685:
7686:
7687:
7688:
7689:
7690:
7691:
7692:
7693:
7694:
7695:
7696:
7697:
7698:
7699:
7700:
7701:
7702:
7703:
7704:
7705:
7706:
7707:
7708:
7709:
7710:
7711:
7712:
7713:
7714:
7715:
7716:
7717:
7718:
7719:
7720:
7721:
7722:
7723:
7724:
7725:
7726:
7727:
7728:
7729:
7730:
7731:
7732:
7733:
7734:
7735:
7736:
7737:
7738:
7739:
7740:
7741:
7742:
7743:
7744:
7745:
7746:
7747:
7748:
7749:
7750:
7751:
7752:
7753:
7754:
7755:
7756:
7757:
7758:
7759:
7760:
7761:
7762:
7763:
7764:
7765:
7766:
7767:
7768:
7769:
7770:
7771:
7772:
7773:
7774:
7775:
7776:
7777:
7778:
7779:
7780:
7781:
7782:
7783:
7784:
7785:
7786:
7787:
7788:
7789:
7790:
7791:
7792:
7793:
7794:
7795:
7796:
7797:
7798:
7799:
7800:
7801:
7802:
7803:
7804:
7805:
7806:
7807:
7808:
7809:
7810:
7811:
7812:
7813:
7814:
7815:
7816:
7817:
7818:
7819:
7820:
7821:
7822:
7823:
7824:
7825:
7826:
7827:
7828:
7829:
7830:
7831:
7832:
7833:
7834:
7835:
7836:
7837:
7838:
7839:
7840:
7841:
7842:
7843:
7844:
7845:
7846:
7847:
7848:
7849:
7850:
7851:
7852:
7853:
7854:
7855:
7856:
7857:
7858:
7859:
7860:
7861:
7862:
7863:
7864:
7865:
7866:
7867:
7868:
7869:
7870:
7871:
7872:
7873:
7874:
7875:
7876:
7877:
7878:
7879:
7880:
7881:
7882:
7883:
7884:
7885:
7886:
7887:
7888:
7889:
7890:
7891:
7892:
7893:
7894:
7895:
7896:
7897:
7898:
7899:
7900:
7901:
7902:
7903:
7904:
7905:
7906:
7907:
7908:
7909:
7910:
7911:
7912:
7913:
7914:
7915:
7916:
7917:
7918:
7919:
7920:
7921:
7922:
7923:
7924:
7925:
7926:
7927:
7928:
7929:
7930:
7931:
7932:
7933:
7934:
7935:
7936:
7937:
7938:
7939:
7940:
7941:
7942:
7943:
7944:
7945:
7946:
7947:
7948:
7949:
7950:
7951:
7952:
7953:
7954:
7955:
7956:
7957:
7958:
7959:
7960:
7961:
7962:
7963:
7964:
7965:
7966:
7967:
7968:
7969:
7970:
7971:
7972:
7973:
7974:
7975:
7976:
7977:
7978:
7979:
7980:
7981:
7982:
7983:
7984:
7985:
7986:
7987:
7988:
7989:
7990:
7991:
7992:
7993:
7994:
7995:
7996:
7997:
7998:
7999:
8000:
8001:
8002:
8003:
8004:
8005:
8006:
8007:
8008:
8009:
8010:
8011:
8012:
8013:
8014:
8015:
8016:
8017:
8018:
8019:
8020:
8021:
8022:
8023:
8024:
8025:
8026:
8027:
8028:
8029:
8030:
8031:
8032:
8033:
8034:
8035:
8036:
8037:
8038:
8039:
8040:
8041:
8042:
8043:
8044:
8045:
8046:
8047:
8048:
8049:
8050:
8051:
8052:
8053:
8054:
8055:
8056:
8057:
8058:
8059:
8060:
8061:
8062:
8063:
8064:
8065:
8066:
8067:
8068:
8069:
8070:
8071:
8072:
8073:
8074:
8075:
8076:
8077:
8078:
8079:
8080:
8081:
8082:
8083:
8084:
8085:
8086:
8087:
8088:
8089:
8090:
8091:
8092:
8093:
8094:
8095:
8096:
8097:
8098:
8099:
8100:
8101:
8102:
8103:
8104:
8105:
8106:
8107:
8108:
8109:
8110:
8111:
8112:
8113:
8114:
8115:
8116:
8117:
8118:
8119:
8120:
8121:
8122:
8123:
8124:
8125:
8126:
8127:
8128:
8129:
8130:
8131:
8132:
8133:
8134:
8135:
8136:
8137:
8138:
8139:
8140:
8141:
8142:
8143:
8144:
8145:
8146:
8147:
8148:
8149:
8150:
8151:
8152:
8153:
8154:
8155:
8156:
8157:
8158:
8159:
8160:
8161:
8162:
8163:
8164:
8165:
8166:
8167:
8168:
8169:
8170:
8171:
8172:
8173:
8174:
8175:
8176:
8177:
8178:
8179:
8180:
8181:
8182:
8183:
8184:
8185:
8186:
8187:
8188:
8189:
8190:
8191:
8192:
8193:
8194:
8195:
8196:
8197:
8198:
8199:
8200:
8201:
8202:
8203:
8204:
8205:
8206:
8207:
8208:
8209:
8210:
8211:
8212:
8213:
8214:
8215:
8216:
8217:
8218:
8219:
8220:
8221:
8222:
8223:
8224:
8225:
8226:
8227:
8228:
8229:
8230:
8231:
8232:
8233:
8234:
8235:
8236:
8237:
8238:
8239:
8240:
8241:
8242:
8243:
8244:
8245:
8246:
8247:
8248:
8249:
8250:
8251:
8252:
8253:
8254:
8255:
8256:
8257:
8258:
8259:
8260:
8261:
8262:
8263:
8264:
8265:
8266:
8267:
8268:
8269:
8270:
8271:
8272:
8273:
8274:
8275:
8276:
8277:
8278:
8279:
8280:
8281:
8282:
8283:
8284:
8285:
8286:
8287:
8288:
8289:
8290:
8291:
8292:
8293:
8294:
8295:
8296:
8297:
8298:
8299:
8300:
8301:
8302:
8303:
8304:
8305:
8306:
8307:
8308:
8309:
8310:
8311:
8312:
8313:
8314:
8315:
8316:
8317:
8318:
8319:
8320:
8321:
8322:
8323:
8324:
8325:
8326:
8327:
8328:
8329:
8330:
8331:
8332:
8333:
8334:
8335:
8336:
8337:
8338:
8339:
8340:
8341:
8342:
8343:
8344:
8345:
8346:
8347:
8348:
8349:
8350:
8351:
8352:
8353:
8354:
8355:
8356:
8357:
8358:
8359:
8360:
8361:
8362:
8363:
8364:
8365:
8366:
8367:
8368:
8369:
8370:
8371:
8372:
8373:
8374:
8375:
8376:
8377:
8378:
8379:
8380:
8381:
8382:
8383:
8384:
8385:
8386:
8387:
8388:
8389:
8390:
8391:
8392:
8393:
8394:
8395:
8396:
8397:
8398:
8399:
8400:
8401:
8402:
8403:
8404:
8405:
8406:
8407:
8408:
8409:
8410:
8411:
8412:
8413:
8414:
8415:
8416:
8417:
8418:
8419:
8420:
8421:
8422:
8423:
8424:
8425:
8426:
8427:
8428:
8429:
8430:
8431:
8432:
8433:
8434:
8435:
8436:
8437:
8438:
8439:
8440:
8441:
8442:
8443:
8444:
8445:
8446:
8447:
8448:
8449:
8450:
8451:
8452:
8453:
8454:
8455:
8456:
8457:
8458:
8459:
8460:
8461:
8462:
8463:
8464:
8465:
8466:
8467:
8468:
8469:
8470:
8471:
8472:
8473:
8474:
8475:
8476:
8477:
8478:
8479:
8480:
8481:
8482:
8483:
8484:
8485:
8486:
8487:
8488:
8489:
8490:
8491:
8492:
8493:
8494:
8495:
8496:
8497:
8498:
8499:
8500:
8501:
8502:
8503:
8504:
8505:
8506:
8507:
8508:
8509:
8510:
8511:
8512:
8513:
8514:
8515:
8516:
8517:
8518:
8519:
8520:
8521:
8522:
8523:
8524:
8525:
8526:
8527:
8528:
8529:
8530:
8531:
8532:
8533:
8534:
8535:
8536:
8537:
8538:
8539:
8540:
8541:
8542:
8543:
8544:
8545:
8546:
8547:
8548:
8549:
8550:
8551:
8552:
8553:
8554:
8555:
8556:
8557:
8558:
8559:
8560:
8561:
8562:
8563:
8564:
8565:
8566:
8567:
8568:
8569:
8570:
8571:
8572:
8573:
8574:
8575:
8576:
8577:
8578:
8579:
8580:
8581:
8582:
8583:
8584:
8585:
8586:
8587:
8588:
8589:
8590:
8591:
8592:
8593:
8594:
8595:
8596:
8597:
8598:
8599:
8600:
8601:
8602:
8603:
8604:
8605:
8606:
8607:
8608:
8609:
8610:
8611:
8612:
8613:
8614:
8615:
8616:
8617:
8618:
8619:
8620:
8621:
8622:
8623:
8624:
8625:
8626:
8627:
8628:
8629:
8630:
8631:
8632:
8633:
8634:
8635:
8636:
8637:
8638:
8639:
8640:
8641:
8642:
8643:
8644:
8645:
8646:
8647:
8648:
8649:
8650:
86
```

GETTING THE LED OUT

In any number of monitoring and/or control uses for your ZX/TS, you may find that using a TV set to check on your system is inconvenient or impractical. This would be so, for example, in outdoor or dirty environments or when you simply wish to check the status of one or two system parameters without the bother of looking up a TV or monitor. One fairly easy and inexpensive way to do this is to use seven-segment LED (light emitting diode) displays as your system output. Your ZX/TS bus doesn't have enough power in reserve to directly drive these displays, but a few simple IC's and an I/O board can give you this capability.

You must have some sort of 8 bit output port. I used Sher-I's Report Generator board, but JK Audio's IIO and even Byte Back's RS-1 module (without the relays) should be useable as a buffered output from your computer. The only other hardware you'll need, aside from miscellaneous wire and some resistors, is a 7415, hex inverting buffer; a 4511 BCD (Binary Coded Decimal)-to-7 segment decoder driver; and a 3 character, 7 segment LED display (MAN-3). A schematic of this simple circuit is given in Fig 1.

This circuit represents a "middle ground" approach to producing a 3 digit output display. More hardware (latches etc.) could have been used, resulting in permanently lit displays, or we could have used software to create a controlling outer loop to keep the monitoring function as the main operating system. The combination of hardware and software I chose represents a system which uses multiplexing to provide the appearance of a full three digit display.

As you can see from Figure one, we are using the ZX's data lines D₀ through D₇ to send out the Binary coded decimal value of one of three digits (units, tens or hundreds). The correct digit is determined by the data lines D₆ through D₈ which complete the path to ground for the appropriate display. Line D₇ enables and latches the last value from lines D₀ to D₃ into the 4511 which in turn, drives the appropriate elements of the display. Multiplexing is provided by the ML software driver routine shown in Fig 2.

After data is POKED into the three buffer locations (Fig 3-a) the ML program outputs the specially constructed character code for each digit in sequence. This operation occurs so rapidly that the eye is fooled into thinking each digit is lit all the time. If you'd like to see what's really happening, you can either increase the dwell time between digits or use the BASIC listing in Fig 3. The BASIC listing requires no machine code but is, of course, slow.

The software is applicable to the Sher-I Report Generator board as it stands. However, you can change the address of the port at 16551(c) to the address of your own OUTPUT subroutine. The Report Generator uses a PEO and is I/O mapped. For memory mapped systems, your OUTPUT subroutine would probably consist of simply putting the value in the accumulator, loading the designated memory location with the value, and returning to the driver.

I built this circuit on a small ACE (all circuit evaluator) board and obtained its power from the +9 volt regulator on the R.O. board. The values for the current limiting resistors are not critical, but should be as large as possible to conserve power, while still giving adequate brightness. Don't overlook the 10K pullup resistors for the 7415, or forget to tie the unused inputs to ground.

Fig 3b gives a BASIC driver which I use to obtain a temperature reading from the A/D converter on the R.O. board and output that value to the LED display. The code shown would actually be a subroutine called from a main menu. The display is normally blank. By touching, say, "T" on my keyboard, I would be selecting a menu item which calls the Temperature output routine. All this can be done without the need of a TV screen, if I know in advance (and I do) which key to press. The system could be expanded, with more chips, lots of software and perhaps even LCD displays to actually let you program your computer without a TV. A consideration here would be to use hex buffer drivers instead of BCD.

Note too, that I used "junk box" parts from around my shop to build this particular display. A proper design would use perhaps a 7412248 for the latch driver (pinout is the same as the 4511). Also remember not to output codes which will turn on all three digits at one time (e.g., code 01111000, produces all 8's), as current drain will be very high.

LEADOUT

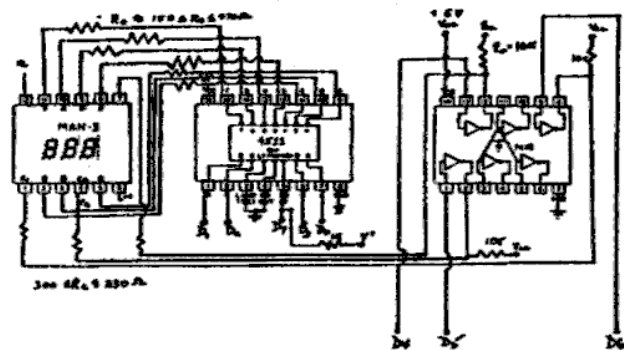


Fig 1 (a)

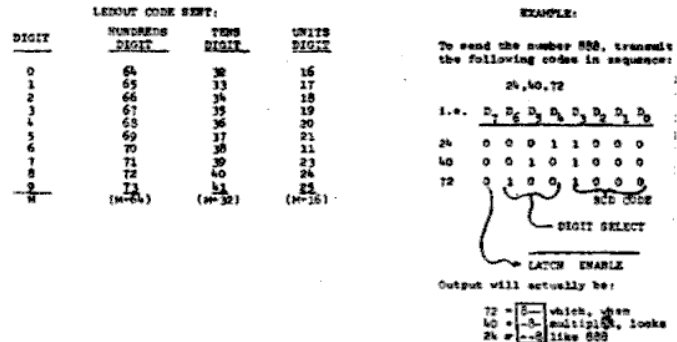


Fig 1 (b)

LEADOUT

NAME	ADDRESS	DATA	COMMENT	CODE
LEADOUT	16540	loop	LEAD,OUT	Load # of Digits
MAIN	16545	LOAD	LEAD,OUT	255740
	16546	LOAD	LEAD,IN	# of Digits to send
DATA	16550	LOAD	LEAD,IN	Save A, Flags
	16551	LOAD	LEAD,IN	Get Value to be output from 4097FF
	16552	LD 4098,A	16551 is poked with value	308240
	16553	POPA	Back to original digit count	71
	16554	INCR	Point to next digit	23
	16555	DECA	Only (A-1) digits left	30
	16556	JUMP 8 Loop	(108A) have we done all 4(5) digits?	0A8A0
	16561	CALL OUTPUT	(2098) send digit out	029820
	16562	CALL DELAY	(40C) leave it lit	00C0A0
	16567	JUMP DATA	(40A) Punctuate each one	03A6A0
	16570	DEC C	Decrement count	00
	16571	BE	We'll be in FF times then return to BASIC	00
	16572	JUMP MAIN	(40A1)	03A1A0
	16573	RETC	Can't get here	00
DELAY	16576	LD DE,00A0	A simple time delay	11A000
	16579	DEC B	For how long we leave 10	10
	16580	JL 40CA	Each digit on	0A8A0
	16583	JUMP 40C3		03C3A0
	16586	LD 40CA,00A0	This is the "outer" loop of the	11A000
	16589	DEC E	Loop of the	10
		BE	Time delay	00
		JUMP 40C3	Keep looping	03C3A0
		RET	can't get here	00

NAME	ADDRESS	DATA	COMMENT
	16535	4097	HUNDREDS DIGIT FROM 6A to 73
	16536	4098	TENS FROM 3E to 41
	16537	4099	UNITS FROM 16 to 25
	16538	409A	BLANKING (255)
	16539, 40, 41	409B, C, D	SPACE

- Fig. 2

ADDRESS: 16534 4096H

16534 26 48 28 18 00 00 FF 00
 16542 01 FF 00 21 97 40 3E 05
 16550 F3 7E 32 82 40 F1 23 30
 16558 CA 8A 40 C0 98 29 CD C0
 16566 40 C3 A6 40 00 C8 C3 A1
 16574 40 C9 11 40 00 1D CA CA
 16582 40 C3 C3 40 11 AA 00 1D
 16590 C8 C3 CD 40 C9

Hex Dump of Leadout
 (Punctuate to be output in BCD)

LIST GROUP



VU-30 OUTPUT - Shaded

MAG=001.83 ROT 347.259 Z=+00270